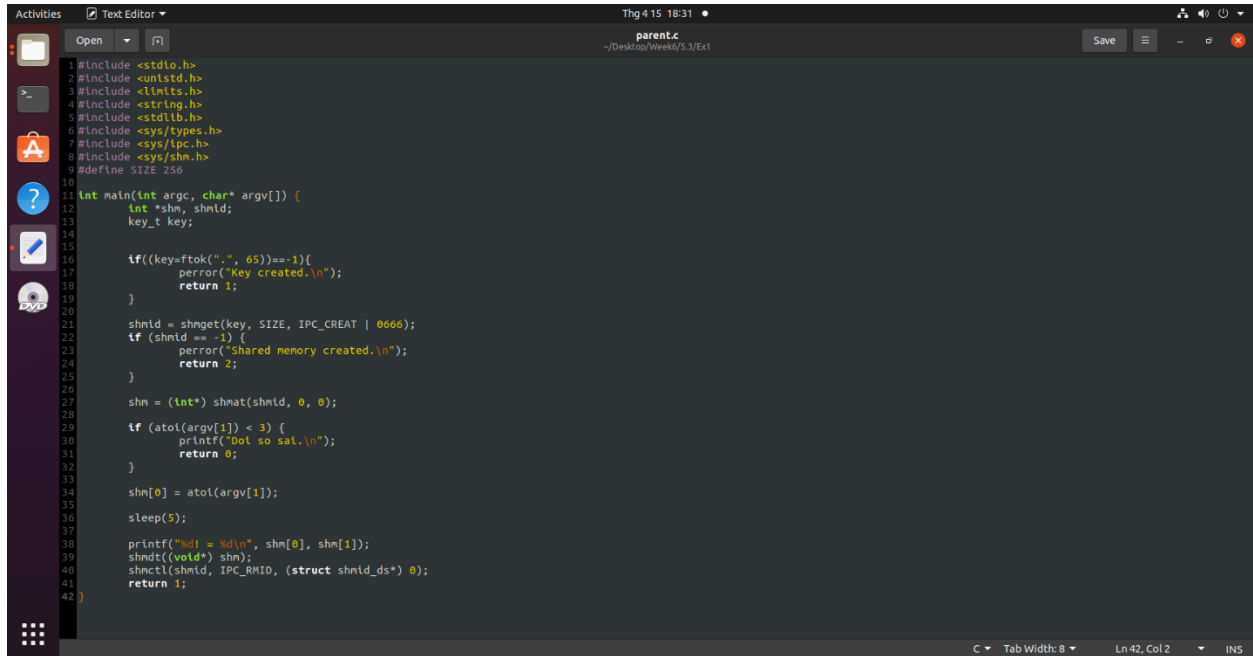


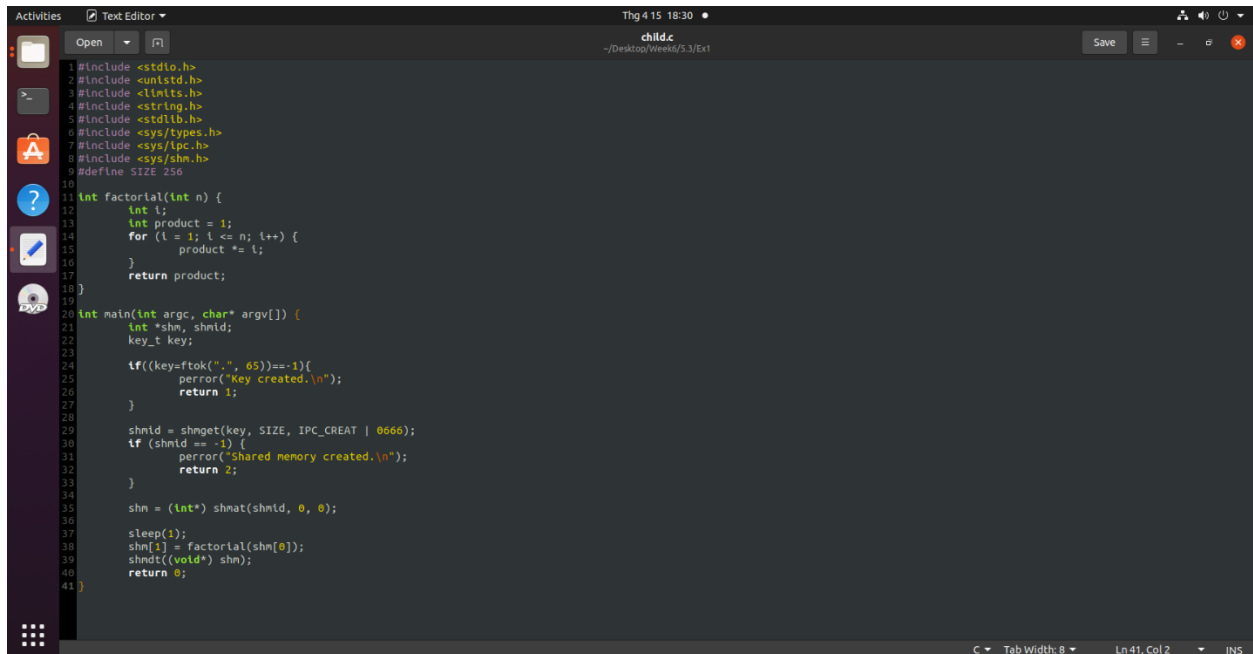
TUẦN 6

5.3 – Vùng nhớ chia sẻ

Bài 1:



```
1#include <stdio.h>
2#include <unistd.h>
3#include <limits.h>
4#include <string.h>
5#include <stdlib.h>
6#include <sys/types.h>
7#include <sys/ipc.h>
8#include <sys/shm.h>
9#define SIZE 256
10
11int main(int argc, char* argv[]) {
12    int *shm, shmid;
13    key_t key;
14
15    if((key=ftok(".", 65))==-1){
16        perror("Key created.\n");
17        return 1;
18    }
19
20    shmid = shmget(key, SIZE, IPC_CREAT | 0666);
21    if (shmid == -1) {
22        perror("Shared memory created.\n");
23        return 2;
24    }
25
26    shm = (int*) shmat(shmid, 0, 0);
27
28    if (atoi(argv[1]) < 3) {
29        printf("Dol so sat.\n");
30        return 0;
31    }
32
33    shm[0] = atoi(argv[1]);
34
35    sleep(5);
36
37    printf("%d = %d\n", shm[0], shm[1]);
38    shmdt((void*) shm);
39    shmctl(shmid, IPC_RMID, (struct shm_id) 0);
40    return 1;
41 }
42 }
```

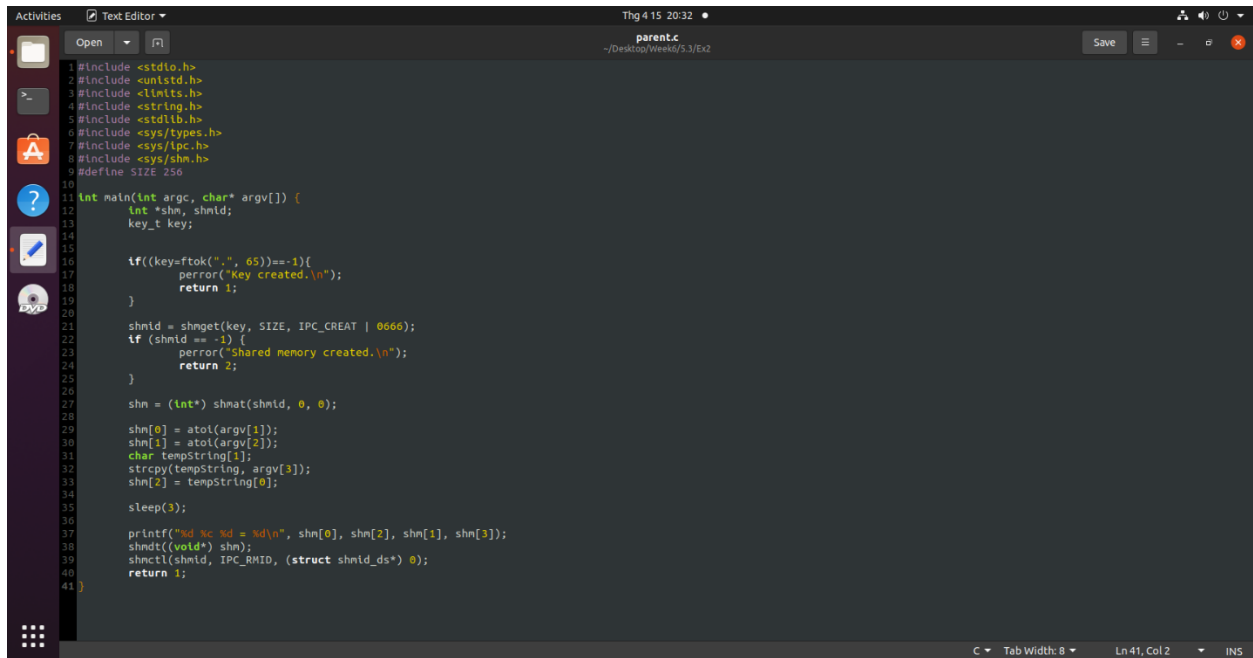


```
1#include <stdio.h>
2#include <unistd.h>
3#include <limits.h>
4#include <string.h>
5#include <stdlib.h>
6#include <sys/types.h>
7#include <sys/ipc.h>
8#include <sys/shm.h>
9#define SIZE 256
10
11int factorial(int n) {
12    int i;
13    int product = 1;
14    for (i = 1; i <= n; i++) {
15        product *= i;
16    }
17    return product;
18 }
19
20int main(int argc, char* argv[]) {
21    int *shm, shmid;
22    key_t key;
23
24    if((key=ftok(".", 65))==-1){
25        perror("Key created.\n");
26        return 1;
27    }
28
29    shmid = shmget(key, SIZE, IPC_CREAT | 0666);
30    if (shmid == -1) {
31        perror("Shared memory created.\n");
32        return 2;
33    }
34
35    shm = (int*) shmat(shmid, 0, 0);
36
37    sleep(1);
38    shm[1] = factorial(shm[0]);
39    shmdt((void*) shm);
40    return 0;
41 }
```

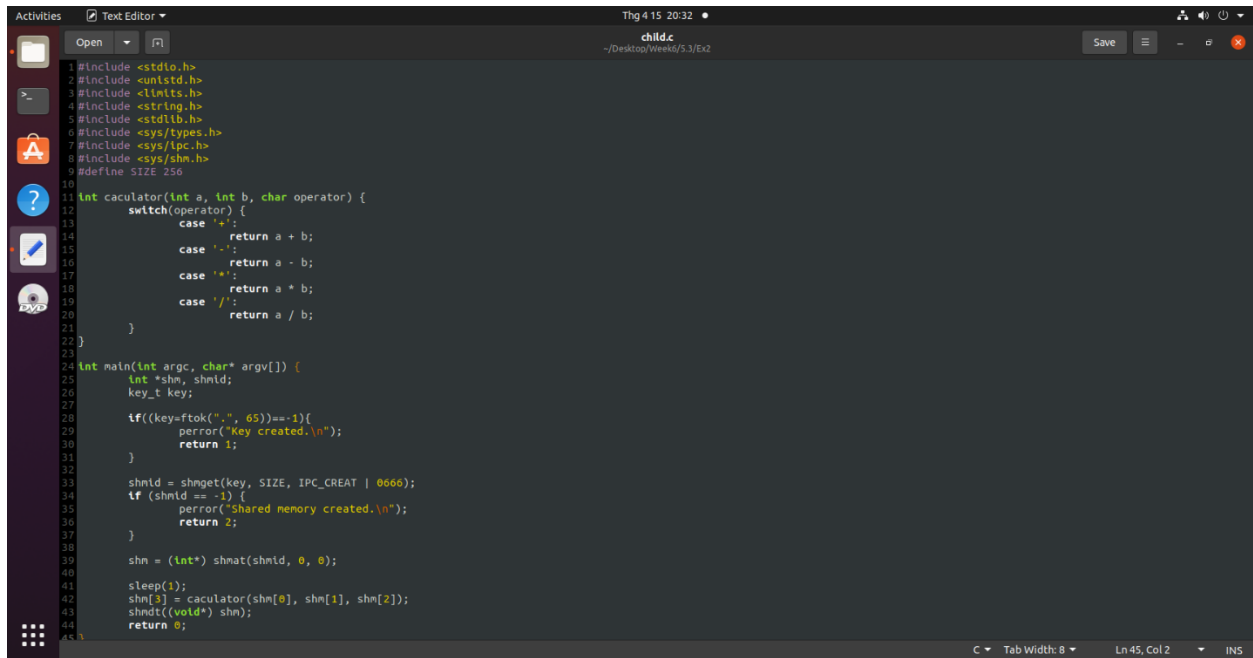
```
Activities Terminal Thg 4 15 18:30
asus@asus-virtual-machine: ~/Desktop/Week6/5.3/Ex1
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ gcc -c parent.c
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ gcc -o parent.out parent.o
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ ./parent.out 5
51 = 120
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$

asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ gcc -c child.c
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ gcc -o child.out child.o
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex1$
```

Bài 2:



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <limits.h>
4 #include <string.h>
5 #include <stdlib.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/shm.h>
9 #define SIZE 256
10
11 int main(int argc, char* argv[]) {
12     int *shm, shmid;
13     key_t key;
14
15     if((key=ftok(".", 65))==-1){
16         perror("key created.\n");
17         return 1;
18     }
19
20     shmid = shmget(key, SIZE, IPC_CREAT | 0666);
21     if (shmid == -1) {
22         perror("Shared memory created.\n");
23         return 2;
24     }
25
26     shm = (int*) shmat(shmid, 0, 0);
27
28     shm[0] = atoi(argv[1]);
29     shm[1] = atoi(argv[2]);
30     char tempString[1];
31     strcpy(tempString, argv[3]);
32     shm[2] = tempString[0];
33
34     sleep(3);
35
36     printf("nd %c %d = %d\n", shm[0], shm[2], shm[1], shm[3]);
37     shmdt((void*) shm);
38     shmctl(shmid, IPC_RMID, (struct shm_id*) 0);
39     return 1;
40 }
41
```



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <limits.h>
4 #include <string.h>
5 #include <stdlib.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/shm.h>
9 #define SIZE 256
10
11 int calculator(int a, int b, char operator) {
12     switch(operator) {
13         case '+':
14             return a + b;
15         case '-':
16             return a - b;
17         case '*':
18             return a * b;
19         case '/':
20             return a / b;
21     }
22 }
23
24 int main(int argc, char* argv[]) {
25     int *shm, shmid;
26     key_t key;
27
28     if((key=ftok(".", 65))==-1){
29         perror("key created.\n");
30         return 1;
31     }
32
33     shmid = shmget(key, SIZE, IPC_CREAT | 0666);
34     if (shmid == -1) {
35         perror("Shared memory created.\n");
36         return 2;
37     }
38
39     shm = (int*) shmat(shmid, 0, 0);
40
41     sleep(1);
42     shm[3] = calculator(shm[0], shm[1], shm[2]);
43     shmdt((void*) shm);
44     return 0;
45 }
```

The image shows two terminal windows side-by-side on a Linux desktop. The desktop background is dark purple. On the left, there is a dock with icons for Activities, Terminal, and several application icons (a file manager, a web browser, and a question mark). The top of the screen shows the system menu with 'Activities', 'Terminal', and the date/time 'Thu 4 15 20:32'.

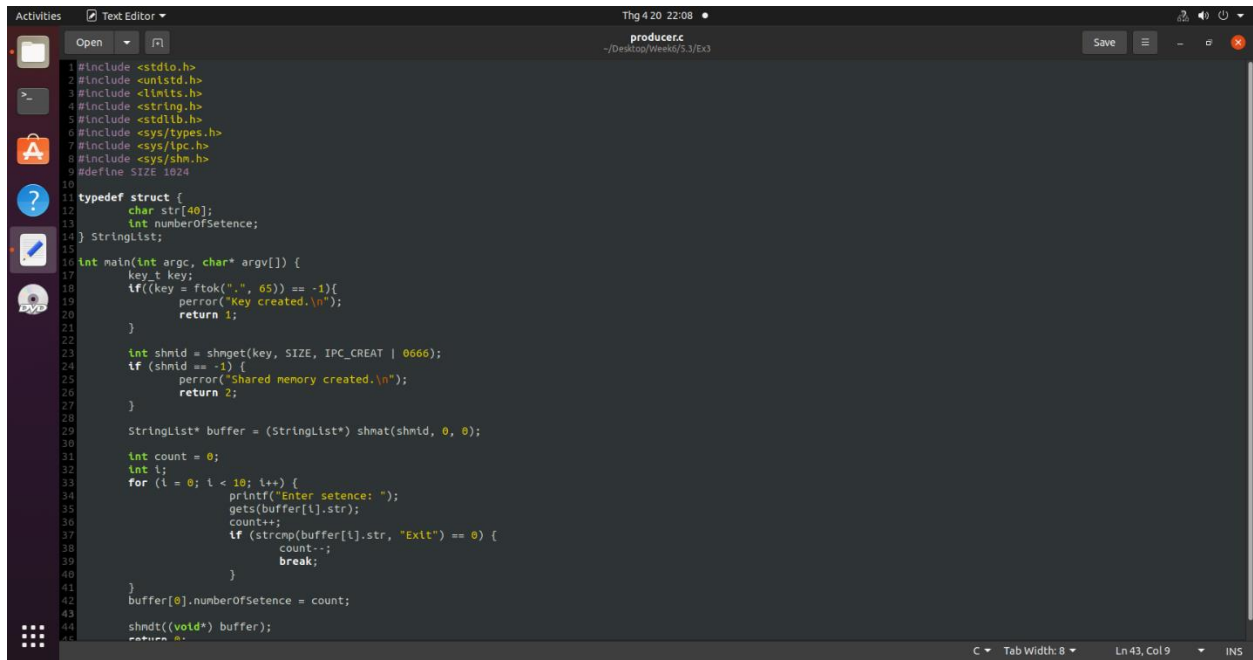
The left terminal window has the title bar 'asus@asus-virtual-machine: ~/Desktop/Week6/5.3/Ex2'. It contains the following commands and output:

```
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ gcc -c parent.c
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ gcc -o parent.out parent.o
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./parent.out 4 6 +
4 + 6 = 10
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./parent.out 4 6 -
4 - 6 = -2
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./parent.out 4 6 *
4 * 6 = 24
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./parent.out 4 6 /
4 / 6 = 0
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$
```

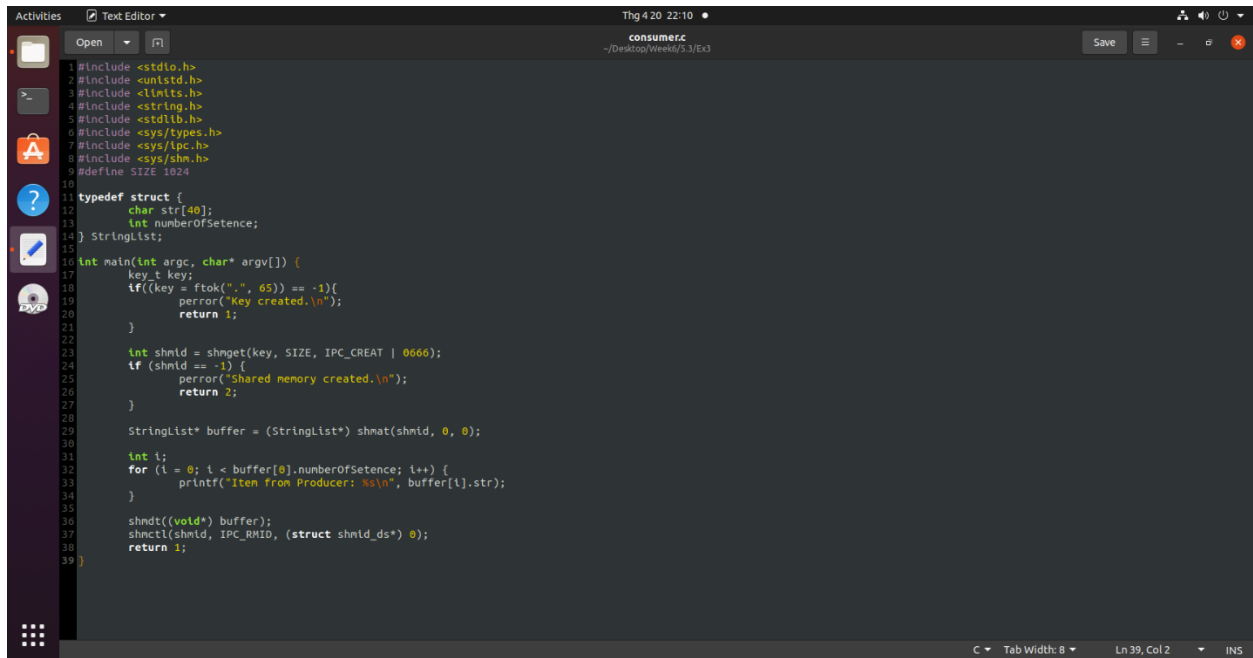
The right terminal window has the title bar 'asus@asus-virtual-machine: ~/Desktop/Week6/5.3/Ex2'. It contains the following commands and output:

```
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ gcc -c child.c
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ gcc -o child.out child.o
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex2$
```

Bài 3:



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <limits.h>
4 #include <string.h>
5 #include <stdlib.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/shm.h>
9 #define SIZE 1024
10
11 typedef struct {
12     char str[40];
13     int numberOfSetence;
14 } StringList;
15
16 int main(int argc, char* argv[]) {
17     key_t key;
18     if((key = ftok(".", 65)) == -1){
19         perror("Key created.\n");
20         return 1;
21     }
22
23     int shmid = shmget(key, SIZE, IPC_CREAT | 0666);
24     if (shmid == -1) {
25         perror("Shared memory created.\n");
26         return 2;
27     }
28
29     StringList* buffer = (StringList*) shmat(shmid, 0, 0);
30
31     int count = 0;
32     int i;
33     for (i = 0; i < 10; i++) {
34         printf("Enter setence: ");
35         gets(buffer[i].str);
36         count++;
37         if (strcmp(buffer[i].str, "Exit") == 0) {
38             count--;
39             break;
40         }
41     }
42     buffer[0].numberOfSetence = count;
43     shmctl((void*) buffer);
44     return 0;
45 }
```

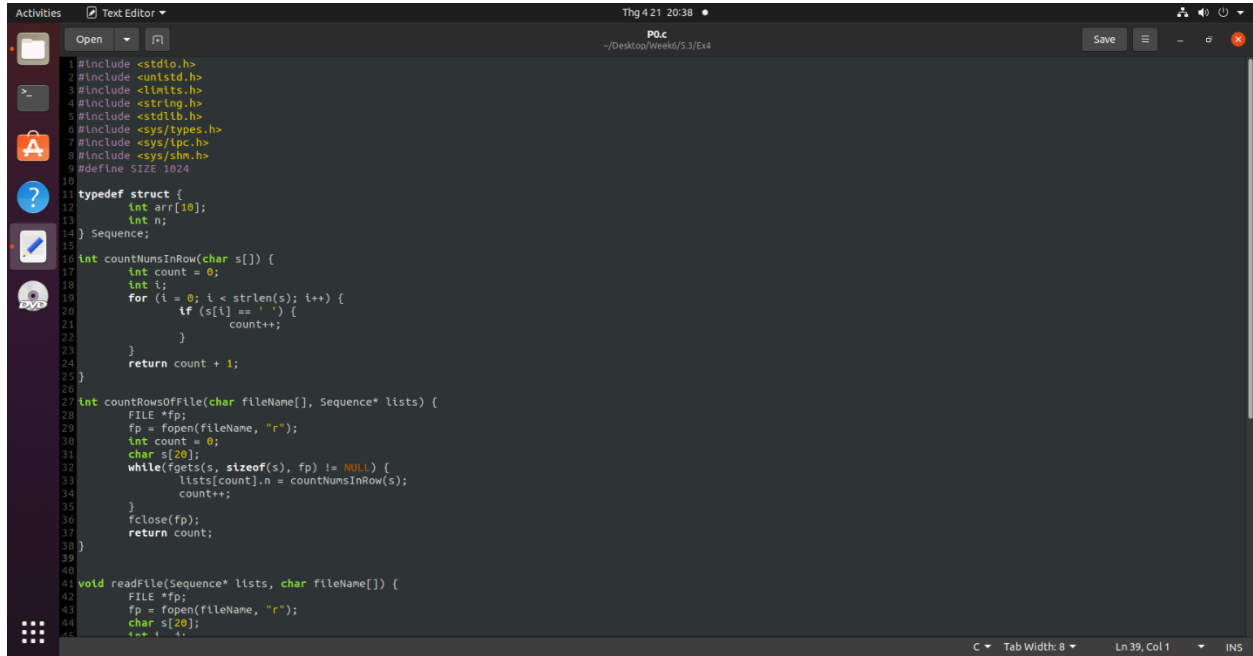


```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <limits.h>
4 #include <string.h>
5 #include <stdlib.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/shm.h>
9 #define SIZE 1024
10
11 typedef struct {
12     char str[40];
13     int numberOfSetence;
14 } StringList;
15
16 int main(int argc, char* argv[]) {
17     key_t key;
18     if((key = ftok(".", 65)) == -1){
19         perror("Key created.\n");
20         return 1;
21     }
22
23     int shmid = shmget(key, SIZE, IPC_CREAT | 0666);
24     if (shmid == -1) {
25         perror("Shared memory created.\n");
26         return 2;
27     }
28
29     StringList* buffer = (StringList*) shmat(shmid, 0, 0);
30
31     int i;
32     for (i = 0; i < buffer[0].numberOfSetence; i++) {
33         printf("Item from Producer: %s\n", buffer[i].str);
34     }
35
36     shmctl((void*) buffer);
37     shmctl(shmid, IPC_RMID, (struct shm_id_s*) 0);
38     return 1;
39 }
```

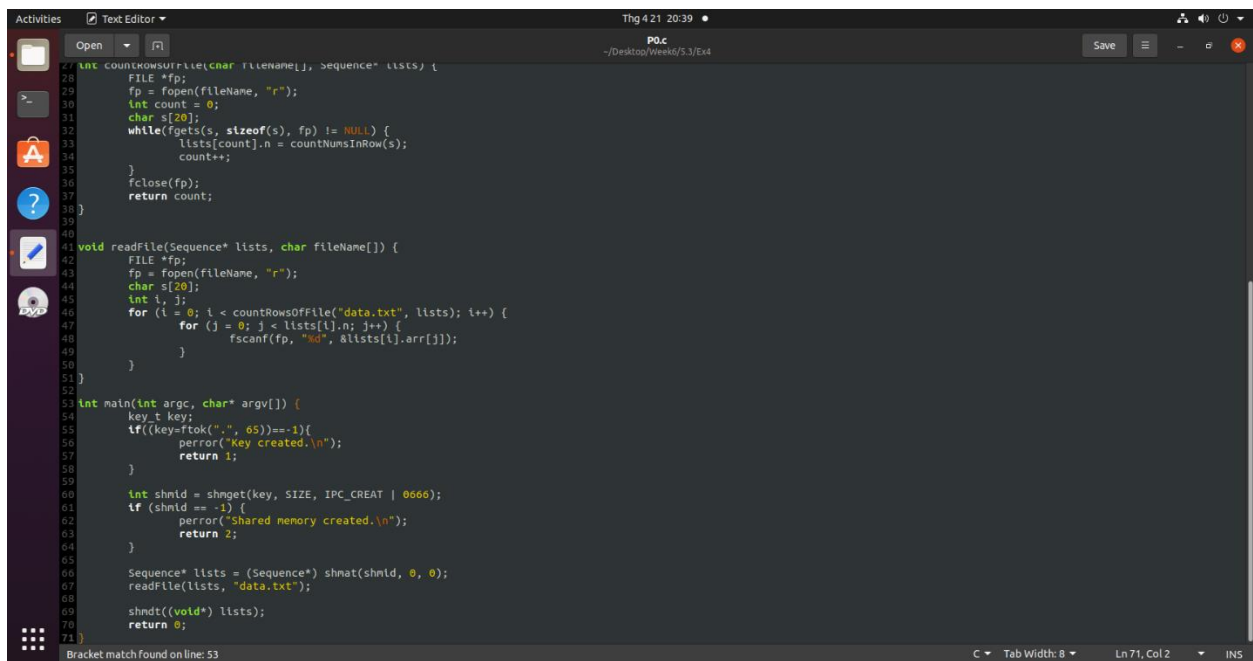
```
Activities Terminal Thg 4 20 22:12
asus@asus-virtual-machine: ~/Desktop/Week6/5.3/Ex3
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ gcc -c producer.c
producer.c: In function 'main':
producer.c:35:4: warning: implicit declaration of function 'gets'; did you mean 'fgets'?
[...Implicit-function-declaration]
35 |     gets(buffer[i].str);
    |     ^~~~~
    |     fgets
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ gcc -o producer.out producer.o
/usr/bin/ld: producer.o: in function 'main':
producer.c:(.text+0xdf): warning: the 'gets' function is dangerous and should not be used
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ ./producer.out
Enter setence: Hello
Enter setence: How
Enter setence: Are
Enter setence: Y
Enter setence: Exit
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ ./producer.out
Enter setence: Hello
Enter setence: How
Enter setence: Are
Enter setence: You
Enter setence: I
Enter setence: Am
Enter setence: Fine
Enter setence: Thanks
Enter setence: And
Enter setence: You?
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$

asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ gcc -c consumer.c
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ gcc -o consumer.out consumer.o
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ ./consumer.out
Item from Producer: Hello
Item from Producer: How
Item from Producer: Are
Item from Producer: Y
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$ ./consumer.out
Item from Producer: Hello
Item from Producer: How
Item from Producer: Are
Item from Producer: You
Item from Producer: I
Item from Producer: Am
Item from Producer: Fine
Item from Producer: Thanks
Item from Producer: And
Item from Producer: You?
asus@asus-virtual-machine:~/Desktop/Week6/5.3/Ex3$
```

Bài 4:



```
1#include <stdio.h>
2#include <unistd.h>
3#include <limits.h>
4#include <string.h>
5#include <stdlib.h>
6#include <sys/types.h>
7#include <sys/ipc.h>
8#include <sys/shm.h>
9#define SIZE 1024
10
11typedef struct {
12    int arr[10];
13    int n;
14} Sequence;
15
16int countNumsInRow(char s[]) {
17    int count = 0;
18    int i;
19    for (i = 0; i < strlen(s); i++) {
20        if (s[i] == ' ') {
21            count++;
22        }
23    }
24    return count + 1;
25}
26
27int countRowsOfFile(char fileName[], Sequence* lists) {
28    FILE *fp;
29    fp = fopen(fileName, "r");
30    int count = 0;
31    char s[20];
32    while (fgets(s, sizeof(s), fp) != NULL) {
33        lists[count].n = countNumsInRow(s);
34        count++;
35    }
36    fclose(fp);
37    return count;
38}
39
40void readFile(Sequence* lists, char fileName[]) {
41    FILE *fp;
42    fp = fopen(fileName, "r");
43    char s[20];
44    while (fgets(s, sizeof(s), fp) != NULL) {
45        // ...
46    }
47    fclose(fp);
48}
```



```
27int countRowsOfFile(char fileName[], Sequence* lists) {
28    FILE *fp;
29    fp = fopen(fileName, "r");
30    int count = 0;
31    char s[20];
32    while (fgets(s, sizeof(s), fp) != NULL) {
33        lists[count].n = countNumsInRow(s);
34        count++;
35    }
36    fclose(fp);
37    return count;
38}
39
40void readFile(Sequence* lists, char fileName[]) {
41    FILE *fp;
42    fp = fopen(fileName, "r");
43    char s[20];
44    while (fgets(s, sizeof(s), fp) != NULL) {
45        // ...
46    }
47    fclose(fp);
48}
49
50int main(int argc, char* argv[]) {
51    key_t key;
52    if ((key = ftok(".", 65)) == -1) {
53        perror("Key created.\n");
54        return 1;
55    }
56    int shmid = shmget(key, SIZE, IPC_CREAT | 0660);
57    if (shmid == -1) {
58        perror("Shared memory created.\n");
59        return 2;
60    }
61    Sequence* lists = (Sequence*) shmat(shmid, 0, 0);
62    readFile(lists, "data.txt");
63    shmdt((void*) lists);
64    return 0;
65}
66
67// ...
68
69// ...
70
71// ...
```

```
Activities Text Editor Thg 4 21 20:39
P1c
~/Desktop/Week6/S.3/Ex4 Save

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <limits.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define SIZE 256

typedef struct {
    int arr[10];
    int n;
} Sequence;

int countRowsOfFile(char fileName[]) {
    FILE *fp;
    fp = fopen(fileName, "r");
    int count = 0;
    char s[20];
    while(fgets(s, sizeof(s), fp) != NULL) {
        count++;
    }
    fclose(fp);
    return count;
}

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void ascendSort(int arr[], int n) {
    int i, j;
    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                swap(&arr[i], &arr[j]);
            }
        }
    }
}

int sumOfNumsInSequence(int arr[], int n) {
    int sum = 0;
    for (i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum;
}

void printArray(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main(int argc, char* argv[]) {
    key_t key;
    if((key=ftok(".", 65))==-1){
        perror("Key created.\n");
        return 1;
    }

    int shmid = shmget(key, SIZE, IPC_CREAT | 0666);
    if (shmid == -1) {
        perror("Shared memory created.\n");
        return 2;
    }

    Sequence* lists = (Sequence*) shmat(shmid, 0, 0);

    sleep(5);
    int i;
    for (i = 0; i < countRowsOfFile("data.txt"); i++) {
        ascendSort(lists[i].arr, lists[i].n);
        printArray(lists[i].arr, lists[i].n);
        printf("Sum = %d\n", sumOfNumsInSequence(lists[i].arr, lists[i].n));
    }

    shmdt((void*) lists);
    shmctl(shmid, IPC_RMID, (struct shm_ds*) 0);
    return 0;
}
```

```
Activities Text Editor Thg 4 21 20:39
P1c
~/Desktop/Week6/S.3/Ex4 Save

int sumOfNumsInSequence(int arr[], int n) {
    int i;
    int sum = 0;
    for (i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum;
}

void printArray(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

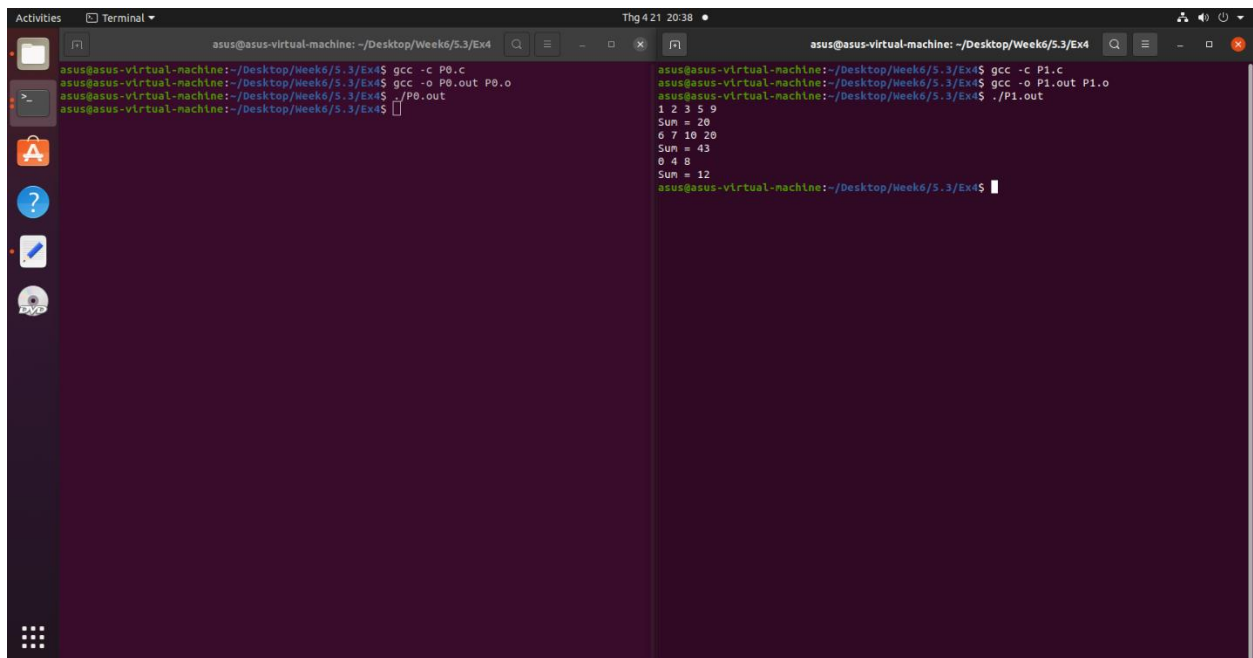
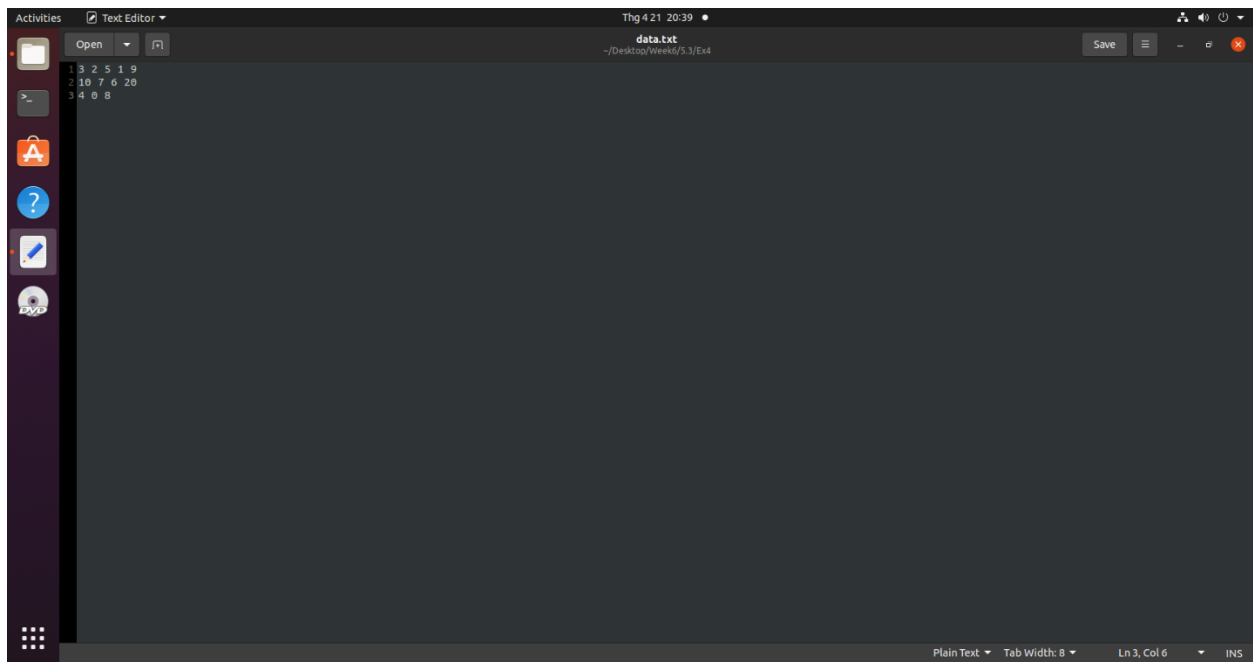
int main(int argc, char* argv[]) {
    key_t key;
    if((key=ftok(".", 65))==-1){
        perror("Key created.\n");
        return 1;
    }

    int shmid = shmget(key, SIZE, IPC_CREAT | 0666);
    if (shmid == -1) {
        perror("Shared memory created.\n");
        return 2;
    }

    Sequence* lists = (Sequence*) shmat(shmid, 0, 0);

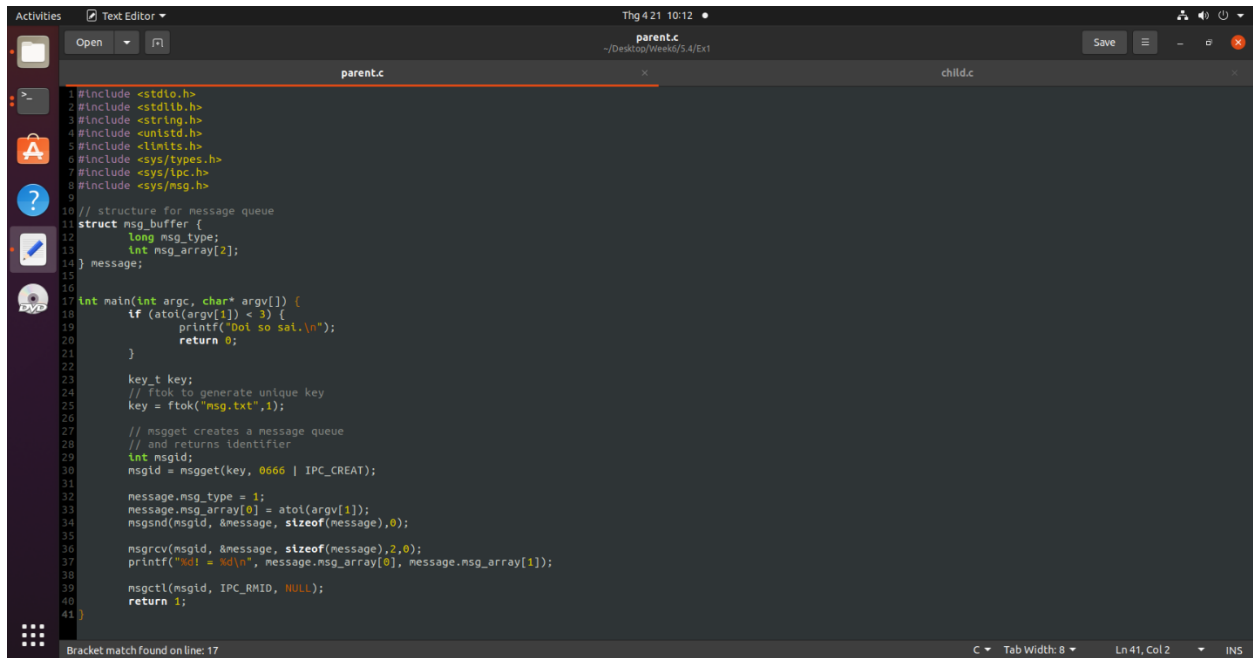
    sleep(5);
    int i;
    for (i = 0; i < countRowsOfFile("data.txt"); i++) {
        ascendSort(lists[i].arr, lists[i].n);
        printArray(lists[i].arr, lists[i].n);
        printf("Sum = %d\n", sumOfNumsInSequence(lists[i].arr, lists[i].n));
    }

    shmdt((void*) lists);
    shmctl(shmid, IPC_RMID, (struct shm_ds*) 0);
    return 0;
}
```

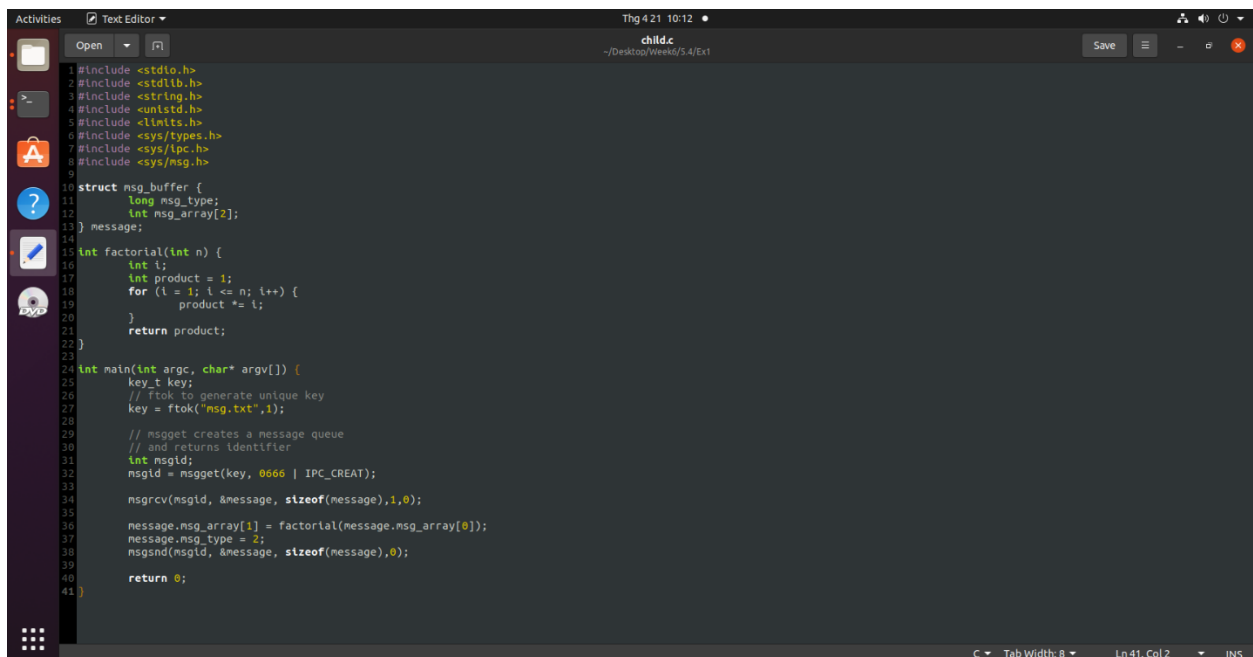
5.4 – Hàng đợi thông điệp

Bài 1:



```
1#include <stdio.h>
2#include <stdlib.h>
3#include <string.h>
4#include <unistd.h>
5#include <limits.h>
6#include <sys/types.h>
7#include <sys/ipc.h>
8#include <sys/msg.h>
9
10// structure for message queue
11struct msg_buffer {
12    long msg_type;
13    int msg_array[2];
14} message;
15
16
17int main(int argc, char* argv[]) {
18    if (atoi(argv[1]) < 3) {
19        printf("Dol so sai.\n");
20        return 0;
21    }
22
23    key_t key;
24    // Ftok to generate unique key
25    key = ftok("msg.txt",1);
26
27    // msgget creates a message queue
28    // and returns identifier
29    int msgid;
30    msgid = msgget(key, 0666 | IPC_CREAT);
31
32    message.msg_type = 1;
33    message.msg_array[0] = atoi(argv[1]);
34    msgsnd(msgid, &message, sizeof(message),0);
35
36    msgrcv(msgid, &message, sizeof(message),2,0);
37    printf("%d\n", message.msg_array[0], message.msg_array[1]);
38
39    msgctl(msgid, IPC_RMID, NULL);
40    return 1;
41}
```

Bracket match found on line: 17



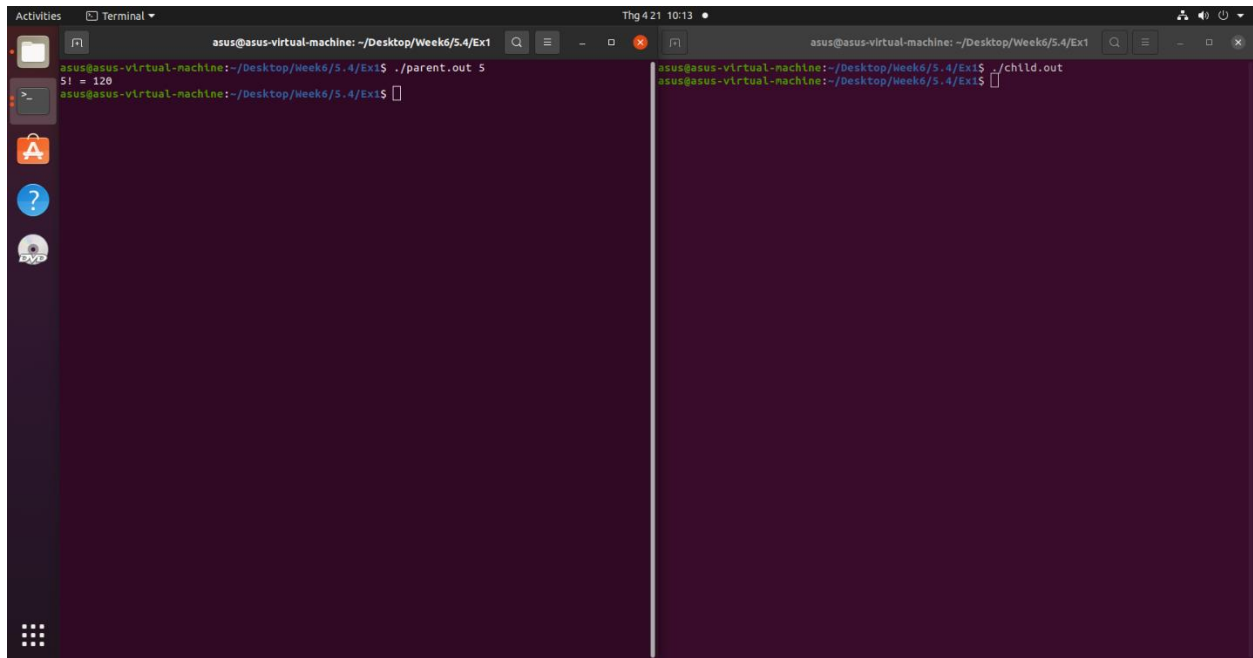
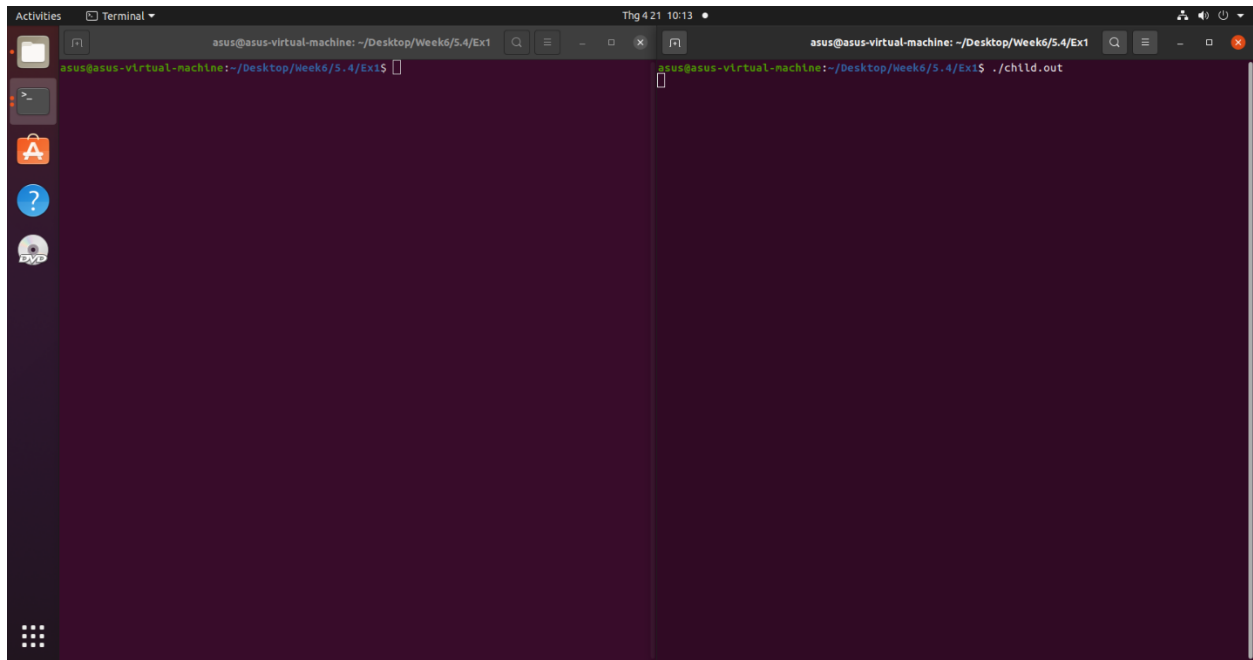
```
1#include <stdio.h>
2#include <stdlib.h>
3#include <string.h>
4#include <unistd.h>
5#include <limits.h>
6#include <sys/types.h>
7#include <sys/ipc.h>
8#include <sys/msg.h>
9
10struct msg_buffer {
11    long msg_type;
12    int msg_array[2];
13} message;
14
15int factorial(int n) {
16    int i;
17    int product = 1;
18    for (i = 1; i <= n; i++) {
19        product *= i;
20    }
21    return product;
22}
23
24int main(int argc, char* argv[]) {
25    key_t key;
26    // Ftok to generate unique key
27    key = ftok("msg.txt",1);
28
29    // msgget creates a message queue
30    // and returns identifier
31    int msgid;
32    msgid = msgget(key, 0666 | IPC_CREAT);
33
34    msgrcv(msgid, &message, sizeof(message),1,0);
35
36    message.msg_array[1] = factorial(message.msg_array[0]);
37    message.msg_type = 2;
38    msgsnd(msgid, &message, sizeof(message),0);
39
40    return 0;
41}
```

```
Activities Terminal Thg 4 21 10:13
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex1
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -c parent.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -o parent.out parent.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ ./parent.out 5

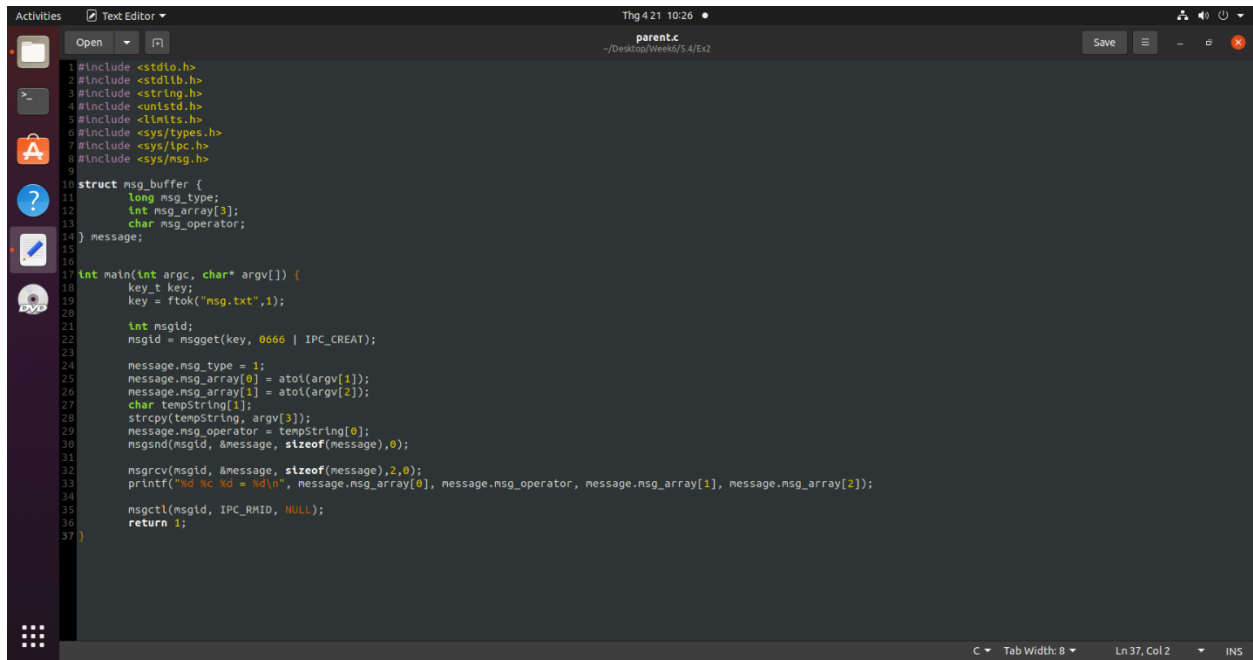
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -c child.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -o child.out child.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$
```

```
Activities Terminal Thg 4 21 10:13
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex1
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -c parent.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -o parent.out parent.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ ./parent.out 5
5! = 120
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$

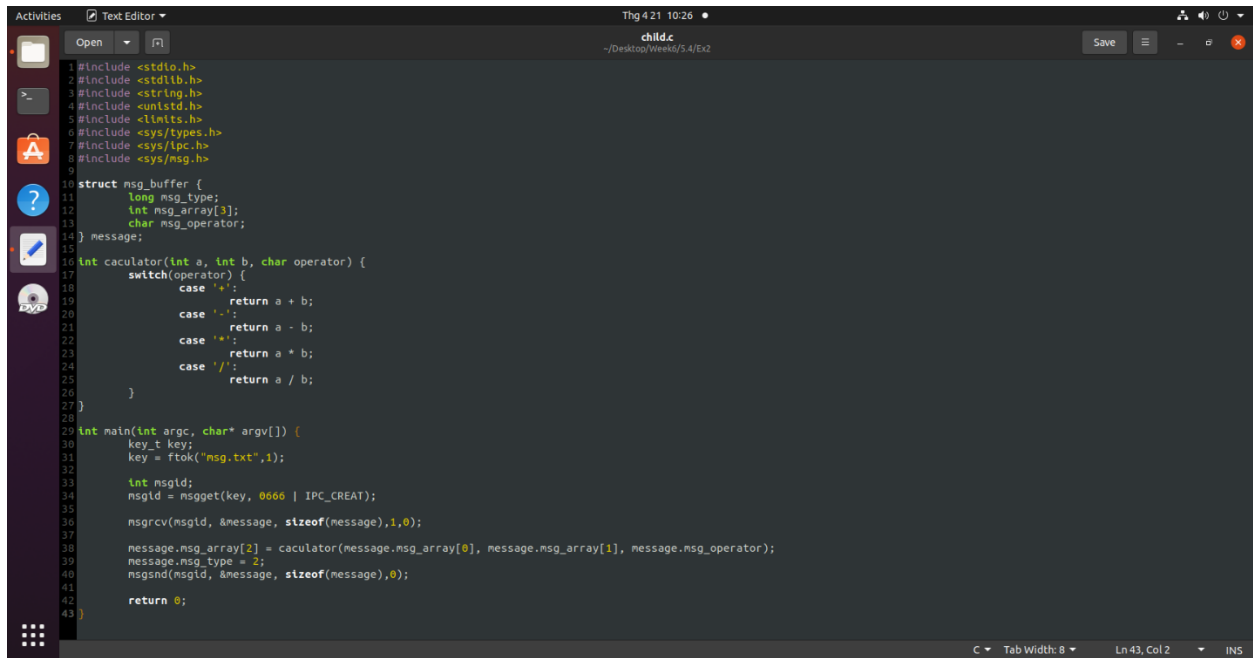
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -c child.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ gcc -o child.out child.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex1$
```



Bài 2:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <limits.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/msg.h>
9
10 struct msg_buffer {
11     long msg_type;
12     int msg_array[3];
13     char msg_operator;
14 } message;
15
16
17 int main(int argc, char* argv[]) {
18     key_t key;
19     key = ftok("msg.txt", 1);
20
21     int msgid;
22     msgid = msgget(key, 0666 | IPC_CREAT);
23
24     message.msg_type = 1;
25     message.msg_array[0] = atoi(argv[1]);
26     message.msg_array[1] = atoi(argv[2]);
27     char tempString[1];
28     strcpy(tempString, argv[3]);
29     message.msg_operator = tempString[0];
30     msgsnd(msgid, &message, sizeof(message), 0);
31
32     msgrcv(msgid, &message, sizeof(message), 2, 0);
33     printf("%d %d %d = %d\n", message.msg_array[0], message.msg_array[1], message.msg_array[2]);
34
35     msgctl(msgid, IPC_RMID, NULL);
36     return 1;
37 }
```



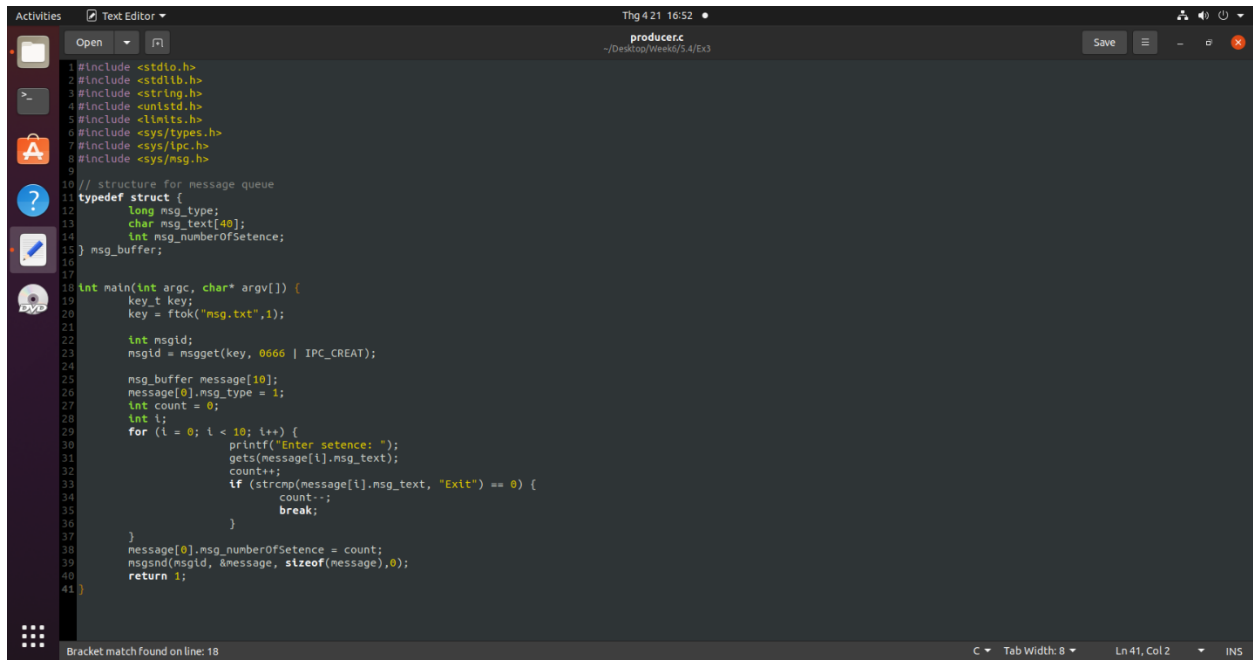
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <limits.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/msg.h>
9
10 struct msg_buffer {
11     long msg_type;
12     int msg_array[3];
13     char msg_operator;
14 } message;
15
16
17 int calculator(int a, int b, char operator) {
18     switch(operator) {
19         case '+':
20             return a + b;
21         case '-':
22             return a - b;
23         case '*':
24             return a * b;
25         case '/':
26             return a / b;
27     }
28 }
29
30 int main(int argc, char* argv[]) {
31     key_t key;
32     key = ftok("msg.txt", 1);
33
34     int msgid;
35     msgid = msgget(key, 0666 | IPC_CREAT);
36
37     msgrcv(msgid, &message, sizeof(message), 1, 0);
38
39     message.msg_array[2] = calculator(message.msg_array[0], message.msg_array[1], message.msg_operator);
40     message.msg_type = 2;
41     msgsnd(msgid, &message, sizeof(message), 0);
42
43     return 0;
44 }
```

The image displays two side-by-side terminal windows on a Linux desktop. The desktop background is a dark purple color. On the left, a vertical sidebar contains icons for 'Activities', 'Terminal', and several application icons (a file manager, a web browser, a question mark, and a DVD icon). The left terminal window has a title bar that reads 'asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex2'. It shows the execution of a C program named 'parent.c' which prints the results of arithmetic operations: $4 + 6 = 10$, $4 - 6 = -2$, $4 * 6 = 24$, and $4 / 6 = 0$. The right terminal window has a title bar that reads 'asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex2'. It shows the execution of a C program named 'child.c' which prints the same arithmetic results. Both windows have a search icon and a menu icon in their title bars.

```
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex2
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ gcc -c parent.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ gcc -o parent.out parent.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./parent.out 4 6 +
4 + 6 = 10
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./parent.out 4 6 -
4 - 6 = -2
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./parent.out 4 6 /*
4 * 6 = 24
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./parent.out 4 6 /
4 / 6 = 0
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$
```

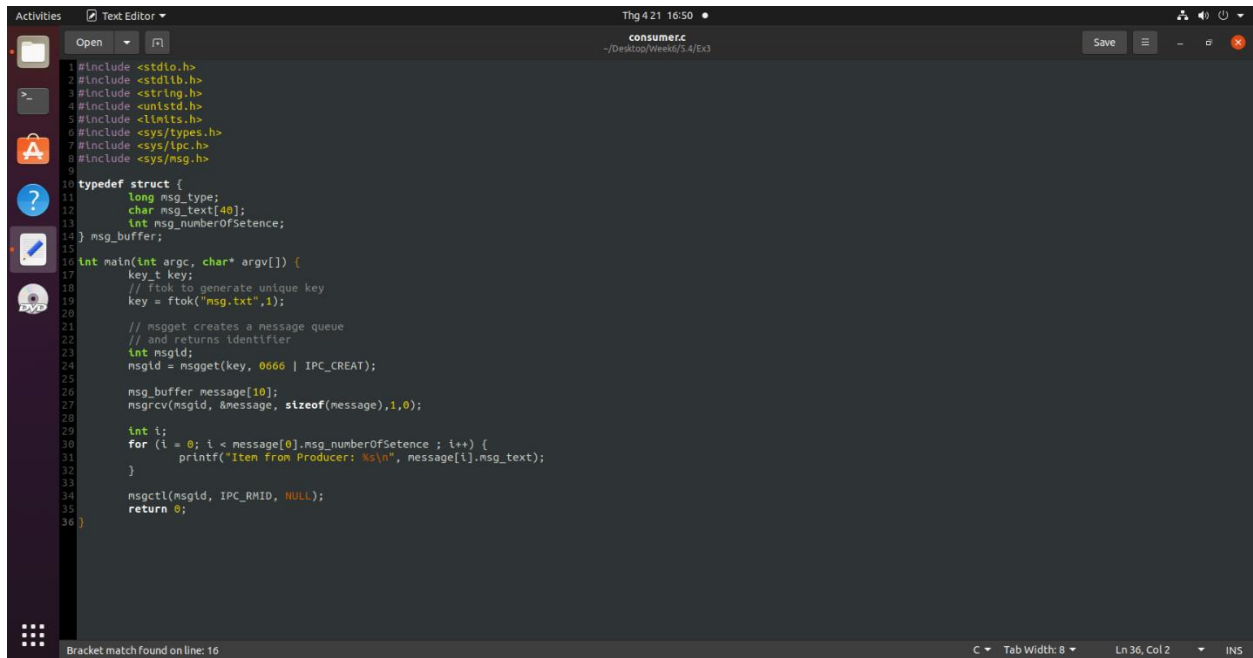
```
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ gcc -c child.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ gcc -o child.out child.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$ ./child.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex2$
```

Bài 3:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <limits.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/msg.h>
9
10 // structure for message queue
11 typedef struct {
12     long msg_type;
13     char msg_text[40];
14     int msg_numberOfSetence;
15 } msg_buffer;
16
17
18 int main(int argc, char* argv[]) {
19     key_t key;
20     key = ftok("msg.txt",1);
21
22     int msgId;
23     msgId = msgget(key, 0666 | IPC_CREAT);
24
25     msg_buffer message[10];
26     message[0].msg_type = 1;
27     int count = 0;
28     int i;
29     for (i = 0; i < 10; i++) {
30         printf("Enter setence: ");
31         gets(message[i].msg_text);
32         count++;
33         if (strcmp(message[i].msg_text, "Exit") == 0) {
34             count--;
35             break;
36         }
37     }
38     message[0].msg_numberOfSetence = count;
39     msgsnd(msgId, &message, sizeof(message),0);
40     return 1;
41 }
```

Bracket match found on line: 18



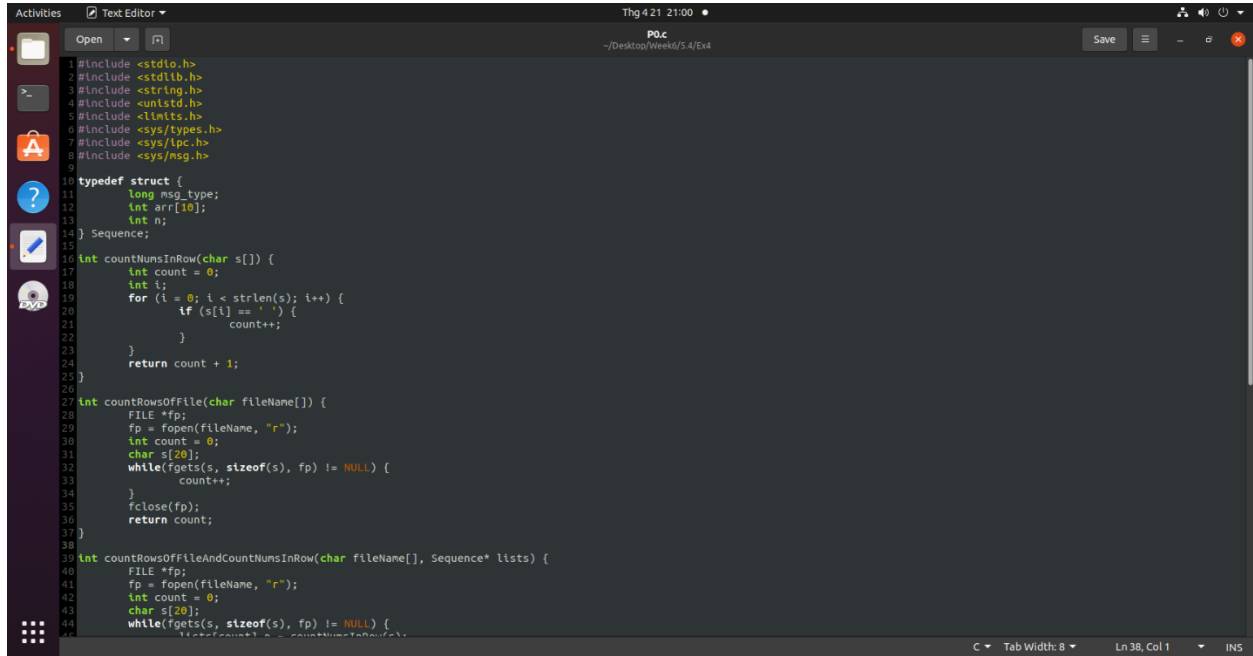
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <limits.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/msg.h>
9
10 typedef struct {
11     long msg_type;
12     char msg_text[40];
13     int msg_numberOfSetence;
14 } msg_buffer;
15
16 int main(int argc, char* argv[]) {
17     key_t key;
18     // ftok to generate unique key
19     key = ftok("msg.txt",1);
20
21     // msgget creates a message queue
22     // and returns identifier
23     int msgId;
24     msgId = msgget(key, 0666 | IPC_CREAT);
25
26     msg_buffer message[10];
27     msgrcv(msgId, &message, sizeof(message),1,0);
28
29     int i;
30     for (i = 0; i < message[0].msg_numberOfSetence ; i++) {
31         printf("Item from Producer: %s\n", message[i].msg_text);
32     }
33
34     msgctl(msgId, IPC_RMID, NULL);
35     return 0;
36 }
```

Bracket match found on line: 16

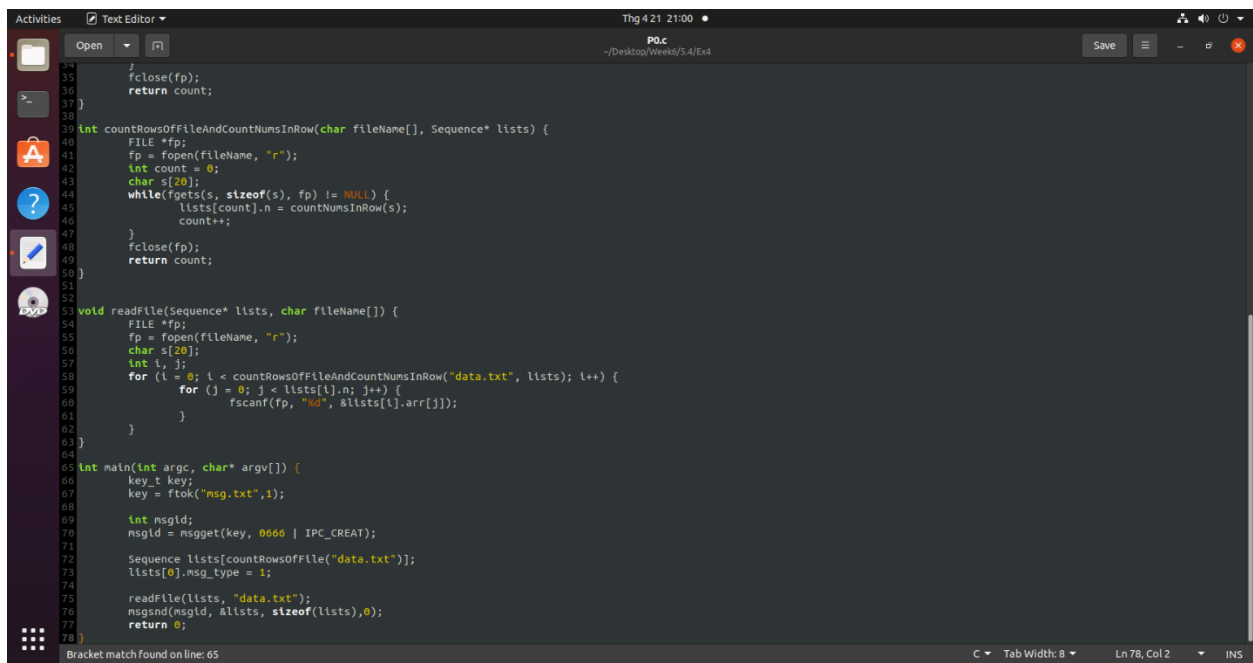
```
Activities Terminal Thg 4 21 16:49
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex3
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ gcc -c producer.c
producer.c: In function 'main':
producer.c:31:4: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
   31 |     gets(message[l].msg_text);
      |     ^~~~~
      |     fgets
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ gcc -o producer.out producer.o
/usr/bin/ld: producer.o: in function 'main':
producer.c:(.text+0x1bd): warning: the 'gets' function is dangerous and should not be used
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ ./producer.out
Enter setence: Hello
Enter setence: How
Enter setence: Are
Enter setence: You
Enter setence: Exit
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ ./producer.out
Enter setence: Hello
Enter setence: How
Enter setence: Are
Enter setence: You
Enter setence: I
Enter setence: Am
Enter setence: Fine
Enter setence: Thanks
Enter setence: And
Enter setence: YOU
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$

asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ gcc -c consumer.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ gcc -o consumer.out consumer.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ ./consumer.out
Item from Producer: Hello
Item from Producer: How
Item from Producer: Are
Item from Producer: You
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$ ./consumer.out
Item from Producer: Hello
Item from Producer: How
Item from Producer: Are
Item from Producer: YOU
Item from Producer: I
Item from Producer: Am
Item from Producer: Fine
Item from Producer: Thanks
Item from Producer: And
Item from Producer: YOU
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex3$
```


Bài 4:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <limits.h>
6 #include <sys/types.h>
7 #include <sys/ipc.h>
8 #include <sys/msg.h>
9
10 typedef struct {
11     long msg_type;
12     int arr[10];
13     int n;
14 } Sequence;
15
16 int countNunsInRow(char s[]) {
17     int count = 0;
18     int i;
19     for (i = 0; i < strlen(s); i++) {
20         if (s[i] == ' ') {
21             count++;
22         }
23     }
24     return count + 1;
25 }
26
27 int countRowsOfFile(char fileName[]) {
28     FILE *fp;
29     fp = fopen(fileName, "r");
30     int count = 0;
31     char s[20];
32     while (fgets(s, sizeof(s), fp) != NULL) {
33         count++;
34     }
35     fclose(fp);
36     return count;
37 }
38
39 int countRowsOfFileAndCountNunsInRow(char fileName[], Sequence* lists) {
40     FILE *fp;
41     fp = fopen(fileName, "r");
42     int count = 0;
43     char s[20];
44     while (fgets(s, sizeof(s), fp) != NULL) {
45         // Here count = countNunsInRow(s);
46     }
```



```
35     }
36     fclose(fp);
37     return count;
38 }
39
40 int countRowsOfFileAndCountNunsInRow(char fileName[], Sequence* lists) {
41     FILE *fp;
42     fp = fopen(fileName, "r");
43     int count = 0;
44     char s[20];
45     while (fgets(s, sizeof(s), fp) != NULL) {
46         lists[count].n = countNunsInRow(s);
47         count++;
48     }
49     fclose(fp);
50     return count;
51 }
52
53 void readFile(Sequence* llists, char fileName[]) {
54     FILE *fp;
55     fp = fopen(fileName, "r");
56     char s[20];
57     int i, j;
58     for (i = 0; i < countRowsOfFileAndCountNunsInRow("data.txt", llists); i++) {
59         for (j = 0; j < llists[i].n; j++) {
60             fscanf(fp, "%d", &lists[i].arr[j]);
61         }
62     }
63 }
64
65 int main(int argc, char* argv[]) {
66     key_t key;
67     key = ftok("msg.txt", 1);
68
69     int msgld;
70     msgld = msgget(key, 0666 | IPC_CREAT);
71
72     Sequence llists[countRowsOfFile("data.txt")];
73     llists[0].msg_type = 1;
74
75     readFile(llists, "data.txt");
76     msgsnd(msgld, &llists, sizeof(llists), 0);
77     return 0;
78 }
```

Bracket match found on line: 65

```
Activities Text Editor Thg 4 21 21:00
P1.c
~/Desktop/Week6/S 4/Ex4 Save

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <limits.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

typedef struct {
    long msg_type;
    int arr[10];
    int n;
} Sequence;

int countRowsOfFile(char fileName[]) {
    FILE *fp;
    fp = fopen(fileName, "r");
    int count = 0;
    char s[20];
    while (fgets(s, sizeof(s), fp) != NULL) {
        count++;
    }
    fclose(fp);
    return count;
}

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void ascendSort(int arr[], int n) {
    int i, j;
    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                swap(&arr[i], &arr[j]);
            }
        }
    }
}

int sumOfNumsInSequence(int arr[], int n) {
    int i;
    int sum = 0;
    for (i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum;
}

int main(int argc, char* argv[]) {
    if (argc < 2) {
        printf("Usage: %s <fileName>\n", argv[0]);
        return 1;
    }
    char fileName[100];
    strcpy(fileName, argv[1]);
    int n = countRowsOfFile(fileName);
    Sequence* lists = malloc(n * sizeof(Sequence));
    int i;
    for (i = 0; i < n; i++) {
        lists[i].msg_type = 1;
        lists[i].n = 0;
        lists[i].arr[0] = 0;
    }
    msgsnd(lists, sizeof(Sequence), 0, 1);
    msgrcv(lists, sizeof(Sequence), 1, 0);
    ascendSort(lists[i].arr, lists[i].n);
    printf("Sum = %d\n", sumOfNumsInSequence(lists[i].arr, lists[i].n));
    msgctl(lists, IPC_RMID, NULL);
    return 0;
}
```

```
Activities Text Editor Thg 4 21 21:00
P1.c
~/Desktop/Week6/S 4/Ex4 Save

    }
    }
}

int sumOfNumsInSequence(int arr[], int n) {
    int i;
    int sum = 0;
    for (i = 0; i < n; i++) {
        sum += arr[i];
    }
    return sum;
}

void printArray(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main(int argc, char* argv[]) {
    key_t key;
    key = ftok("msg.txt", 1);

    int msgId;
    msgId = msgget(key, 0666 | IPC_CREAT);
    Sequence lists[countRowsOfFile("data.txt")];
    msgrcv(msgId, &lists, sizeof(lists), 1, 0);
    int i;
    for (i = 0; i < countRowsOfFile("data.txt"); i++) {
        ascendSort(lists[i].arr, lists[i].n);
        printArray(lists[i].arr, lists[i].n);
        printf("Sum = %d\n", sumOfNumsInSequence(lists[i].arr, lists[i].n));
    }
    msgctl(msgId, IPC_RMID, NULL);
    return 0;
}
```

Bracket match found on line: 62

```
Activities Terminal Thg 4 21 20:59
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex4
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -c P0.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -o P0.out P0.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P0.out
1 2 3 5 9
Sun = 20
6 7 10 20
Sun = 43
0 4 8
Sun = 12
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$
```

```
Activities Terminal Thg 4 21 20:59
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex4
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -c P0.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -o P0.out P0.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P0.out
1 2 3 5 9
Sun = 20
6 7 10 20
Sun = 43
0 4 8
Sun = 12
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P1.out
1 2 3 5 9
Sun = 20
6 7 10 20
Sun = 43
0 4 8
Sun = 12
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$
```

```
Activities Terminal Thg 4 21 20:59
asus@asus-virtual-machine: ~/Desktop/Week6/5.4/Ex4
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -c P0.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -o P0.out P0.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P0.out
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -c P1.c
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ gcc -o P1.out P1.o
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P1.out
1 2 3 5 9
Sum = 20
0 7 10 20
Sum = 43
0 4 8
Sum = 12
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$ ./P1.out
1 2 3 5 9
Sum = 20
0 7 10 20
Sum = 43
0 4 8
Sum = 12
asus@asus-virtual-machine:~/Desktop/Week6/5.4/Ex4$
```

```
Activities Text Editor Thg 4 21 21:00
data.txt
~/Desktop/Week6/5.4/Ex4
1 3 2 5 1 9
2 10 7 6 20
3 4 0 8
Loading file "/home/asus/Desktop/Week6/5.4/Ex4/data.txt"...
```